Index

1. Introduction

2. Learning Finite-State Automata

3. ***Learning Finite-State Transducers***

   - Transduction tasks and Models
   - Subsequential Transducer Learning: OSTIA
   - Using Input/Output syntactic restrictions: OSTIA–DR

4. Applications

Pattern Recognition, Natural Language Processing and Formal Transduction

- (Stochastic) Grammars and Automata are adequate models for *Classification tasks*. But there are many Pattern Recognition problems which are better framed within the most general *Interpretation* paradigm
- Interpretation tasks and be conceptually (and practically) tackled through *Formal Transduction*.
- E.G.: many *Continuous Speech Recognition and Understanding* tasks can be seen as (simple) *transductions* from certain acoustic, phonetic or lexical input sequences into output sequences of higher-level linguistic categories
- Many *direct* applications such as *Language Translation* and *Semantic Decoding*
- *Simple transducers* are often powerful enough to deal with useful mappings between *complex languages*

Index

1. Introduction

2. Learning Finite-State Automata

3. Learning Finite-State Transducers

   - ***Transduction tasks and Models***
   - Subsequential Transducer Learning: OSTIA
   - Using Input/Output syntactic restrictions: OSTIA–DR

4. Applications

Not all the Transduction Tasks are Equally Difficult: Examples

- **1... Spanish to English, word by word**
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  to what time departs the flight more early of Boston to Denver in TWA?

- **2... Division by 7**
  3 5 7 6 8 1 8 0 3 1 (: 7 =)
  0 5 1 0 9 7 4 0 0 4

- **3... English to Decimal**
  NINEHUNDREDANDNINETEENTHOUSANDANDNINE
  9 19 0 09

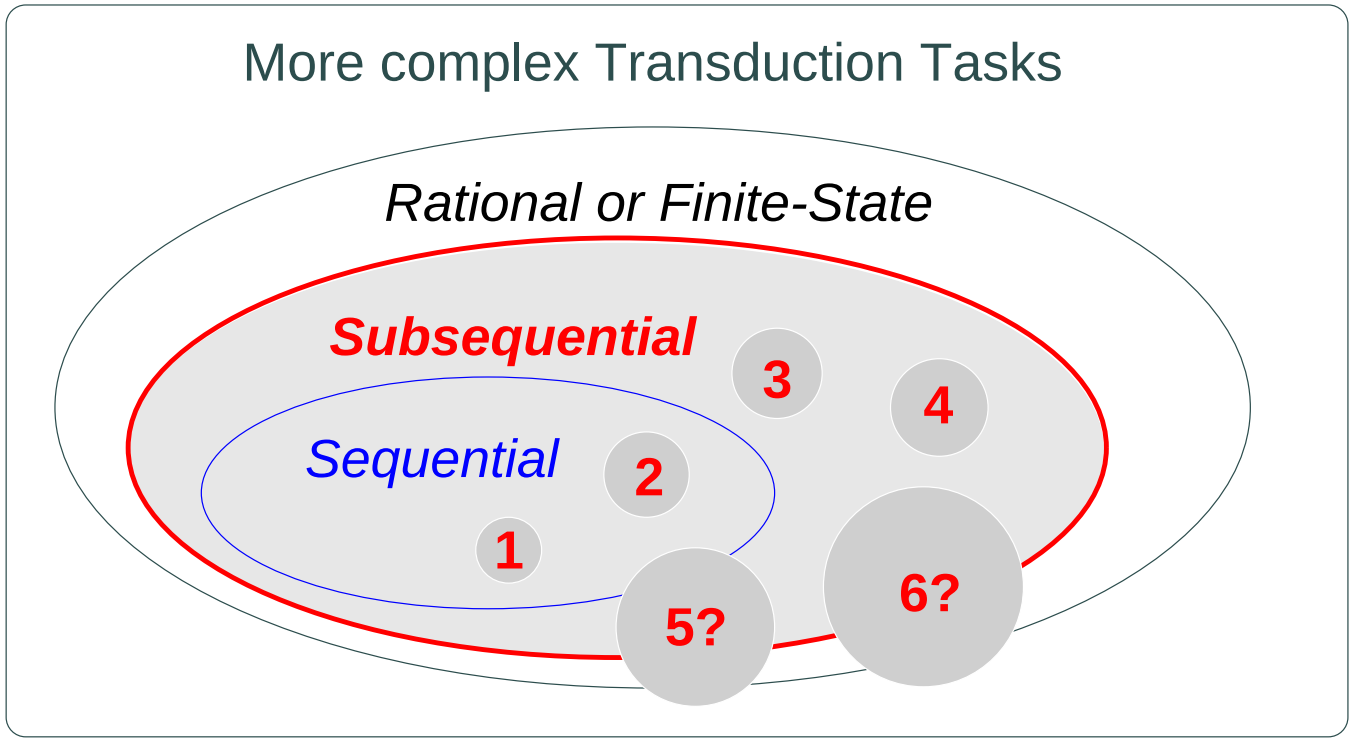
- **4... Roman to Decimal**
  III XIX XLII LXXIV CDII CMLXXXIX
  3 19 4 2 74 4 02 9 8 9

- **5... ATIS: English to ”Pseudo English”**
  WHAT IS THE DEPARTURE TIME OF TWA EARLIEST FLIGHT FROM BOSTON TO DENVER?
  List departure time of earliest morning TWA flights from Boston and to Denver

- **6... Spanish to English**
  ¿A QUE HORA SALE EL VUELO MAS TEMPRANO DE BOSTON A DENVER EN TWA?
  What is the departure time of TWA earliest flight from Boston to Denver?

Not all the Transduction Tasks are Equally Difficult

1. Spanish to English, word by word
2. Division by 7
3. English to Decimal
4. Roman to Decimal
5. ATIS: English to ”Pseudo English”
6. Spanish to English



THE MAIN CONCERN IS THE REQUIRED **degree of “sequentiality”** OR **position monotonicity** BETWEEN INPUT-OUTPUT SUBSEQUENCES

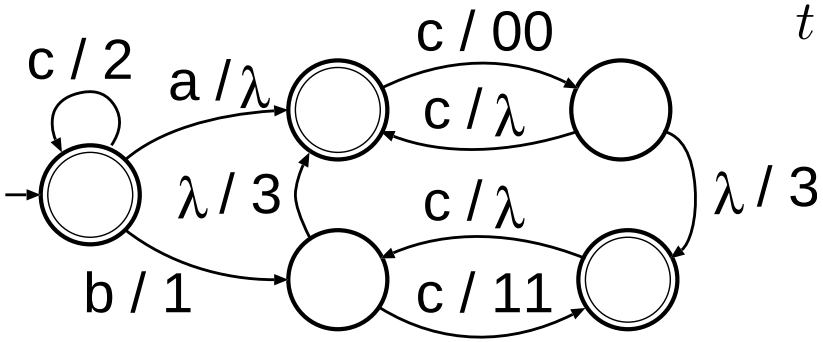
Finite State Transducers: Formal Definition

A *Finite State* or *Rational Transducer* τ is a 6-tuple $\tau = (Q, X, Y, q_0, Q_F, E)$:

Q:	Finite set of *States*
X, Y:	Input and output *Alphabets*
$q_0 \in Q$:	*Initial State*
$Q_F \subset Q$:	Set of *Final States*
$E \subset Q \times X^* \times Y^* \times Q$:	*"Edges"* or *Transitions*

Transitions can equivalently defined as $E \subset Q \times (X \cup \lambda) \times Y^* \times Q$.

EXAMPLE



$t = \{$ (λ, λ), (*cb*, 213), (*ccb*, 2213),
(a, λ), (*ac*, 003), (*cac*, 2003),
(*c*, 2), (*bc*, 111), (*cbc*, 2111),
(*b*, 13), (*bc*, 13003), (*cbc*, 213003),
(*ca*, 2), (*cca*, 22), (*bcc*, 1113),
(*cc*, 22), (*ccc*, 222), ... $\}$

Finite State Transducer Learning and Grammatical Inference

- A Finite State (regular) Grammar (FSG), G, can be seen as a particular case of Finite State Transducer (FST), T which, for each input string x, produces an output string y, such that $y =$YES if x belongs to the language of G and $y =$NO otherwise.

- Any algorithm that would learn any FST can also learn any FSG and, therefore, learning Finite State Transducers (FST) is at least as hard as learning Finite State (regular) Grammars (FSG).

- Transducer Learning can be properly framed within the paradigm of *Grammatical Inference*

Transducer Identification in the Limit:

Let $f : X^* \rightarrow Y^*$ be a transduction function. A transducer learning algorithm $\mathcal{A}$ is said to *identify f in the limit* if, for any positive presentation S of input-output pairs of f, $\mathcal{A}$ converges to a transduction $g : X^* \rightarrow Y^*$ such that $\forall x \in Dom(f), g(x) = f(x)$, when the number of pairs in S tends to infinity.

Sequential Transducers

A *Sequential Transducer* (ST) τ is a 5-tuple $\tau = (Q, X, Y, q_0, E)$:

Q:	Finite set of *States*
X, Y:	Input and output *Alphabets*
$q_0 \in Q$:	*Initial State*
$E \subset Q \times X \times Y^* \times Q$:	*"Edges"* or *Transitions*

- All the states are *accepting*
- Edges are *deterministic*:
  $(q, a, u, r), (q, a, v, s) \in E \Rightarrow (u = v \wedge r = s)$

PROPERTIES:

1. STs $\equiv$ *Generalized Sequential Machines* $\supset$ (Mealy *and* Moore machines)
2. STs *preserve prefixes*: $t(\lambda) = \lambda;\ \ t(uv) \in t(u)Y^*$

"Property" 2 entails *strict sequentiality*,
which can hardly be adequate in many cases of interest

Subsequential Transduction

[Berstel,79]

A *Subsequential Transducer* (SST) τ is a 6-tuple $\tau = (Q, X, Y, q_0, E, \sigma)$, where:

- $\tau' = (Q, X, Y, q_0, E)$ is a Sequential Transducer
- $\sigma : Q \rightarrow Y^*$ is a *state output* (partial) *function*
- For each input string x, the output string y is obtained by concatenating $\sigma(q)$ to $\tau'(x)$, where q is the last state reached through the analysis of x by τ'; i.e.:

$$y = \tau(x) = \tau'(x)\sigma(q)$$

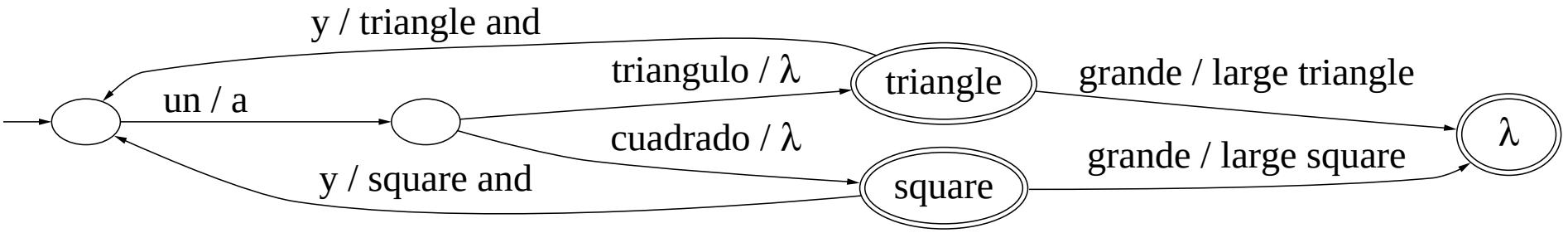
PROPERTIES:

1. Finite State $\supset$ ***Subsequential Transduction*** $\supset$ Sequential.
2. Input-output monotonicity (sequentiality) needs *not* be as strict as in STs.

Subsequential Transducers (intuitive concept)

- Deterministic Finite State networks.
- Accept sentences from an *input* language and produces sentences of an *output* language.
- SST operation relies on "*delaying*" the production of output words until enough of the input sentence has been seen to guarantee a correct output.

An example of SST:



Index

1. Introduction

2. Learning Finite-State Automata

3. Learning Finite-State Transducers

   - Transduction tasks and Models
   - ***Subsequential Transducer Learning: OSTIA***
   - Using Input/Output syntactic restrictions: OSTIA–DR

4. Applications

Learning SSTs: the OSTI Algorithm

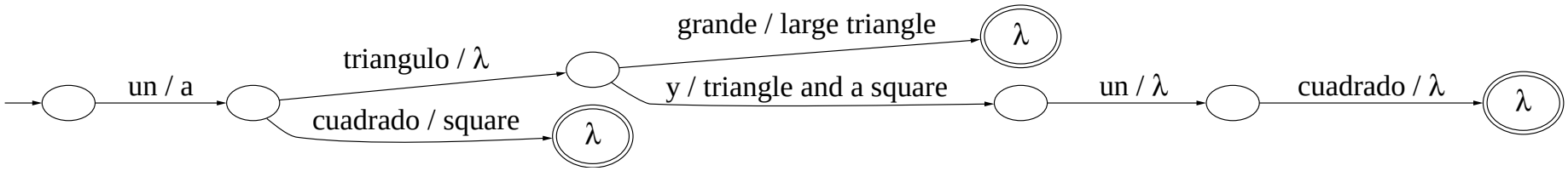
[Oncina, 91-93]

SSTs can be learned from training examples using the
Onward Subsequential Transducer Inference Algorithm (OSTIA).

1. Build an "*onward*" tree representation of the training data (a tree in which output strings are as close as possible to the root – called "OTSST".)

Example:

(*un triángulo y un cuadrado* , a triangle and a square),
(*un triángulo grande* , a large triangle),
(*un cuadrado* , a square)



2. Orderly traverse the tree, joining states to attempt getting adequate generalizations.

OSTIA State-Merging Learning Procedure

- The traversal of the tree goes in a level by level manner, typically by using a lexicographical order of state names.

- Two kinds of State Merging:

  – Merging based on *Local conditions*: involve only the two states under consideration. Most basic basic idea [Oncina, 91-93]:

    If both candidate states have the same output, or at least one has no output, merging is allowed.

  – *Derived merges*: once two states are merged, others may also need to be recursively merged in order to *preserve determinism*.

    "Pushing-back" certain output substrings may be needed in this process.

- If a cascade of derived merges *fails* preserving determinism, the original and all the derived *merges are discarded*

The Onward Subsequential Transducer Inference Algorithm (OSTIA)

Algorithm OSTIA
INPUT: finite set of input-output pairs, $T \subset (X^* \times Y^*)$
OUTPUT: OST τ consistent with T
τ:=OTST(T); q:=$first(\tau)$;
while q $<$ $last(\tau)$ **do** q:=$next(\tau$,q); q':=$first(\tau)$;
while q' $<$ q **do**
if σ(q') $=$ σ(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$ **then** τ':=τ;
$merge(\tau$,q',q);
while $\neg subsequential(\tau)$ **do**
let (r,a,v,s), (r,a,v',s') be two edges of τ that
violate the $subsequential$ condition, with s' $<$ s;
if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile endif**
u:=lcp(v',v);
$push\_back(\tau$,u^{-1}v', (r,a,v',s'));
$push\_back(\tau$,u^{-1}v, (r,a,v,s));
if σ(s') $=$ σ(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
then $merge(\tau$,s',s)
else exitwhile endif
endwhile //$\neg subsequential(\tau)$//
if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile endif**
endif //σ(q') $=$ σ(q)//
q':=$next(\tau$,q');
endwhile //q' $<$ q//
endwhile //q $<$ $last(\tau)$//
end //OSTIA//

Properties of OSTIA learning

[Oncina, García & Vidal, 93]

- *Correctness*: the resulting transducer is *subsequential* and is a (state-merging) *generalization* of the set of training pairs T.
- *Convergence*: Using OSTIA the class of *total* Subsequential Transductions can be *identified in the limit*.
- *Efficiency*: OSTIA average running time is observed to be $O(n(m+k))$, where
  - $n = \sum_{(x,y)\in T} |x|$, (overall length of input strings)
  - $m = \max_{(x,y)\in T} |x|$ (longest output string)
  - $k = |X|$ (size of input alphabet).

  $\Rightarrow$ *huge sets of training examples can be easily handled.*

Applications of SSTs and OSTIA learning

- *Learning several toy but not trivial transduction tasks* [Oncina, 91-93].
  - Simple Arithmetic (e.g., decimal division by a fixed number).
  - Conversion of (large) English Numbers into Decimal notation.
  - Translation of (large) English Numbers into Spanish (and vice versa).
  - Conversion of Roman Numbers into Decimal.
  - etc.

- *Semantic Decoding*:
  - MLA [Castellanos et al.,98]
  - (Subset of) ATIS [Vidal,94]

- *Language Translation*:
  - MLA [Castellanos et al.,94]
  - Traveler Task [Amengual et al., 95-99]

Language Understanding through Semantic Decoding

Given a speech or text input sentence, produce an output which can be used to *drive the actions* specified in this sentence.

TYPICAL EXAMPLES:

- ATIS (Air Travel Information Systems [Price, 90]):
  - **input**: Spontaneous English Sentences
  - **output**: Formal Query commands to the ATIS Data Base
- BDGEO (Spanish Geographic Quest [Diaz, 93]):
  - **input**: Natural Language Spanish Sentences
  - **output**: Formal Query commands to BDGEO
- MLA ("Miniatue Language Acquisition [Feldman et al., 90]):
  - **input**: Quasi-natural English Sentences
  - **output**: First-Order Predicate Logic Formulae

A Simple Experimental Language Understanding Task: MLA

[Feldman et al., 90]

- Involves description and manipulation of simple visual scenes.
- Originally introduced as a challenging Language Learning task with a fairly simple syntax and small lexicon (about 30 words).
- Extended, as required, to study the impact of *increasing complexity, vocabulary size*, etc.

Examples:

a medium light square and a circle are far above a light circle and a medium square

a large dark triangle is added far to the left of the square and the medium circle

the large circle which is above the square and the medium triangle is removed

MLA: Language Understanding through Semantic Decoding

[Castellanos et al.,94-98]

- Visual scenes of MLA "understood" in terms of *(first-order) logic formulae.*
- Objects $=$ Variables: x, y, z, w (allow up to *four* objects in a scene).
- 8 unary predicates on variables for *shape, shade* and *size*
- 9 (0-ary or binary) predicates for object relative positions (*above, below, far below, to the right, touch*, etc).
- Three increasingly non-monotone representations for object relations: L1, L2, L3. Translation into L1 is purely *sequential*; *subsequential* for L2 and L3

Examples:

a small triangle touches a medium light circle and a large square

L1: (Sm(x) & T(x)) **To** (M(z) & Li(z) & C(z) & La(w) & S(w))
L2: Sm(x) & T(x) & **To(x,z)** & M(z) & Li(z) & C(z) & **To(x,w)** & La(w) & S(w)
L3: Sm(x) & T(x) & M(z) & Li(z) & C(z) & La(w) & S(w) & **To(x,z) & To(x,w)**

Air Travel Information System (ATIS): Semantic Decoding

Translate English sentences into a semantic representation in terms of *"Pseudo English" (PE) formal queries*.

EXAMPLES:

---

show all flights and fares from <city> to <∗city>

LIST FLIGHTS FROM <CITY> AND TO <∗CITY> ALONG WITH FARES

I'd like information on <airline> flight from <city> to <∗city>

LIST FLIGHTS FROM <CITY> AND TO <∗CITY> AND <AIRLINE>

I'd like to find chipest fare one-way fare from <city> to <∗city>

LIST CHEAPEST ONE-WAY FARES CHARGED FOR FLIGHTS FROM <CITY> AND TO <∗CITY>

please tell me about ground transportation from <city> airport to downtown <∗city>

LIST GROUND SERVICES PROVIDED FOR <AIRPORT> AND PROVIDED FOR <∗CITY>

what airline is <airline> abbreviation for

LIST AIRLINES WHOSE AIRLINE CODE IS <AIRLINE>

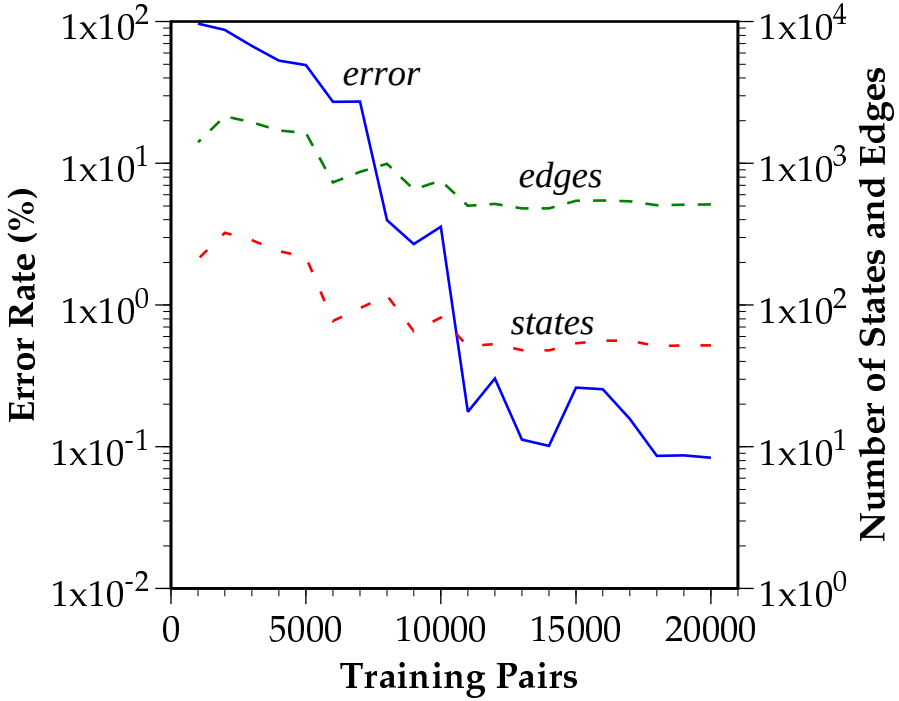
English sentences in lowercase, Pseudo-English commands in capitals.

Tokens within angular brackets are "generic non-terminals" or *bilingual categories*.

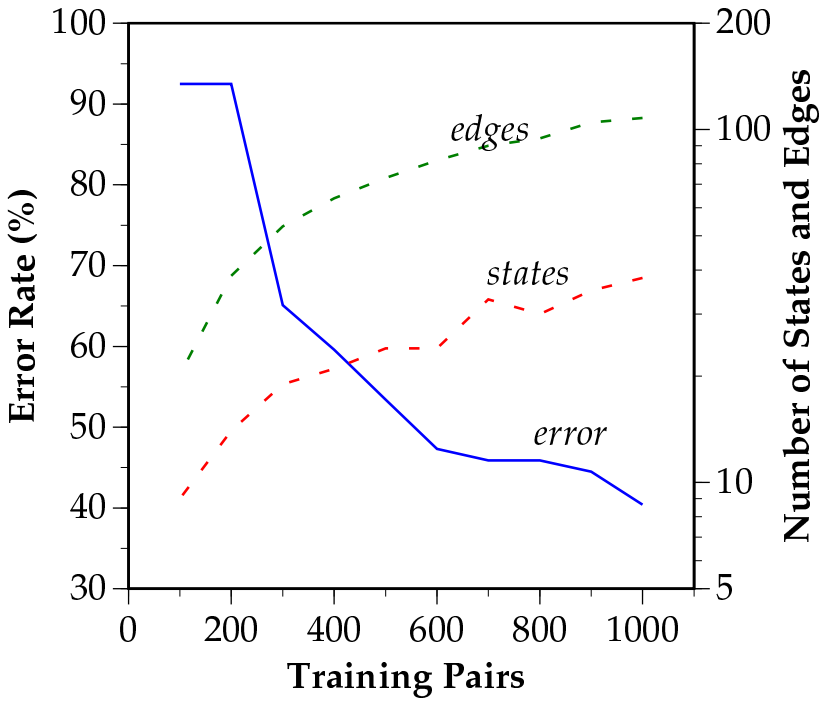
SEMANTIC DECODING: OSTIA LEARNING RESULTS

Evolution of test-set semantic error and size of the OSTIA learned transducers for increasing amounts of training data.

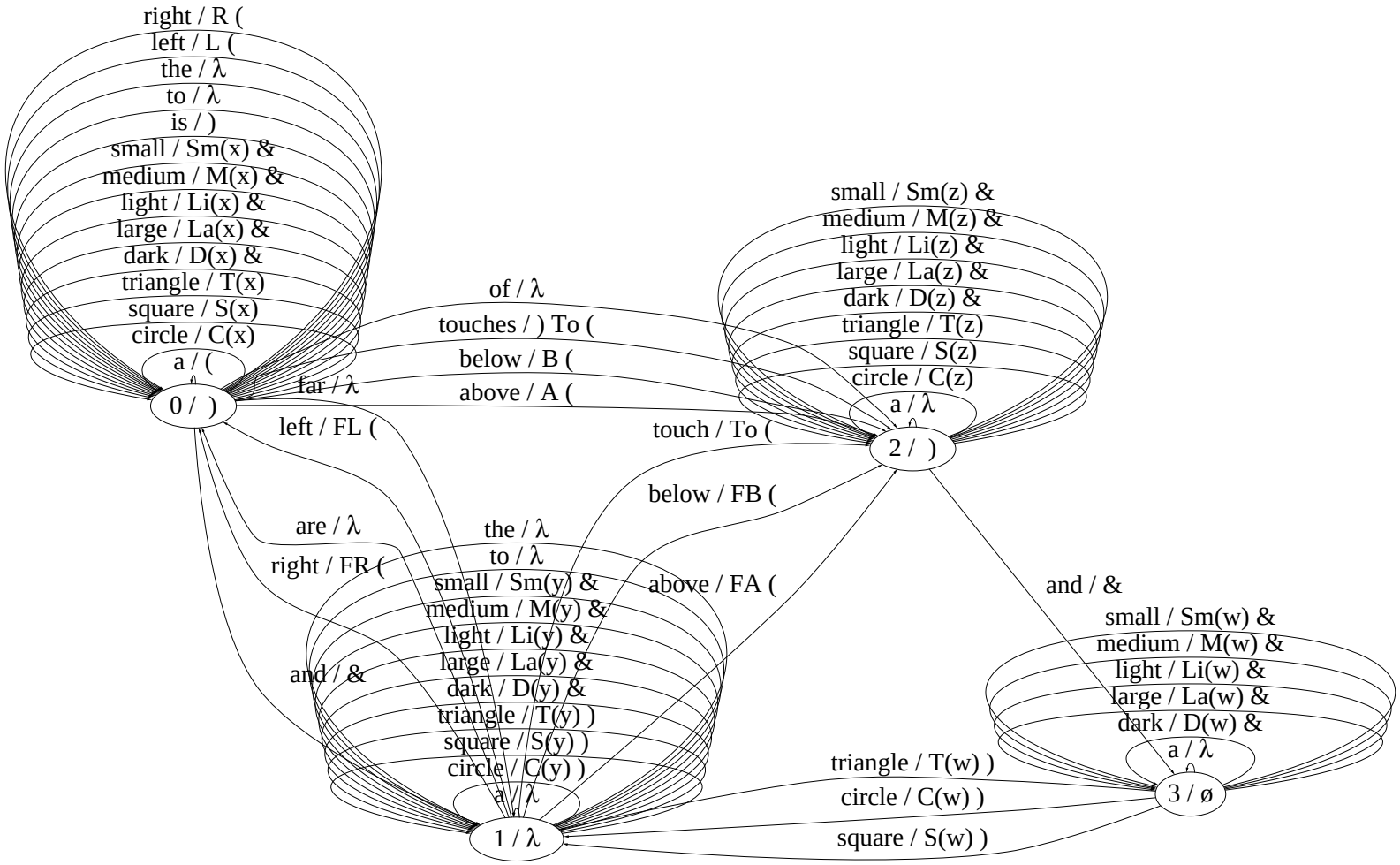
MLA-L3 (10k test sentences)
(similar results for L2;
slightly better for L1)



ATIS (146 test sentences)
(small subset of short,
class A sentences)



OSTIA-learned SST for the MLA Language Understanding Task



Machine Translation (MT) and Subsequential Tansduction

- Translation between languages can be modeled by a Finite State (FS) mappings
- An important advantage of FS Translation Models is their great adequacy to be used for speech-input MT
- Theoretically speaking, most language pairs involve only subsequential mappings (*output text can be produced without having to wait until the end of the input discourse!*)
- In practice, many language pairs do involve only short-term *input/output asyncronies*
- ***Subsequential Transducers*** can be quite appropriate for ***Limited Domain applications***

A Simple Experimental Machine Translation Task: MTA

[Feldman et al., 90] [Castellanos et al., 94]

- Based on MLA (description and manipulation of simple visual scenes), winch was originally introduced as a challenging Language Learning task with a fairly simple syntax and small lexicon (about 30 words).
- Reformulated for Machine Translation and *extended*, as required, to study the impact of increasing degree of input-output *non-monotonicity, vocabulary size*, etc.

Examples:

Spanish:	*un cuadrado mediano y claro y un círculo tocan a un círculo claro y un cuadrado mediano*
English:	a medium light square and a circle touch a light circle and a medium square
Spanish:	*se elimina el círculo grande que esta encima del cuadrado y del triángulo mediano*
English:	the large circle which is above the square and the medium triangle is removed

MTA Translation Results using OSTIA

[Castellanos, Galiano and Vidal, ICGI–94], [Oncina et al., ICSNLP–94]

Spanish-English Translation Word Error Rates (TWER) for the Extended MTA Task, as a function of the Training Set size supplied to OSTIA.
Test Set: 10,000 independent text input sentences.

Train. Size	TWER	States	Edges
1,000	**58.8%**	412	1652
2,000	**57.0%**	846	3197
4,000	**51.8%**	1598	5970
8,000	**3.4%**	186	891
16,000	**0.0%**	17	206

- *Convergence starts from 4,000–8,000 training pairs (decreasing size of the learned transducers).*
- Good results achieved with very compact transducers learned from reasonably small training sets.

▷ Bad news: These SSTs perform very poorly with *imperfect text* or *speech* input.

"Good" basic SSTs can accept incorrect input producing even more incorrect output!

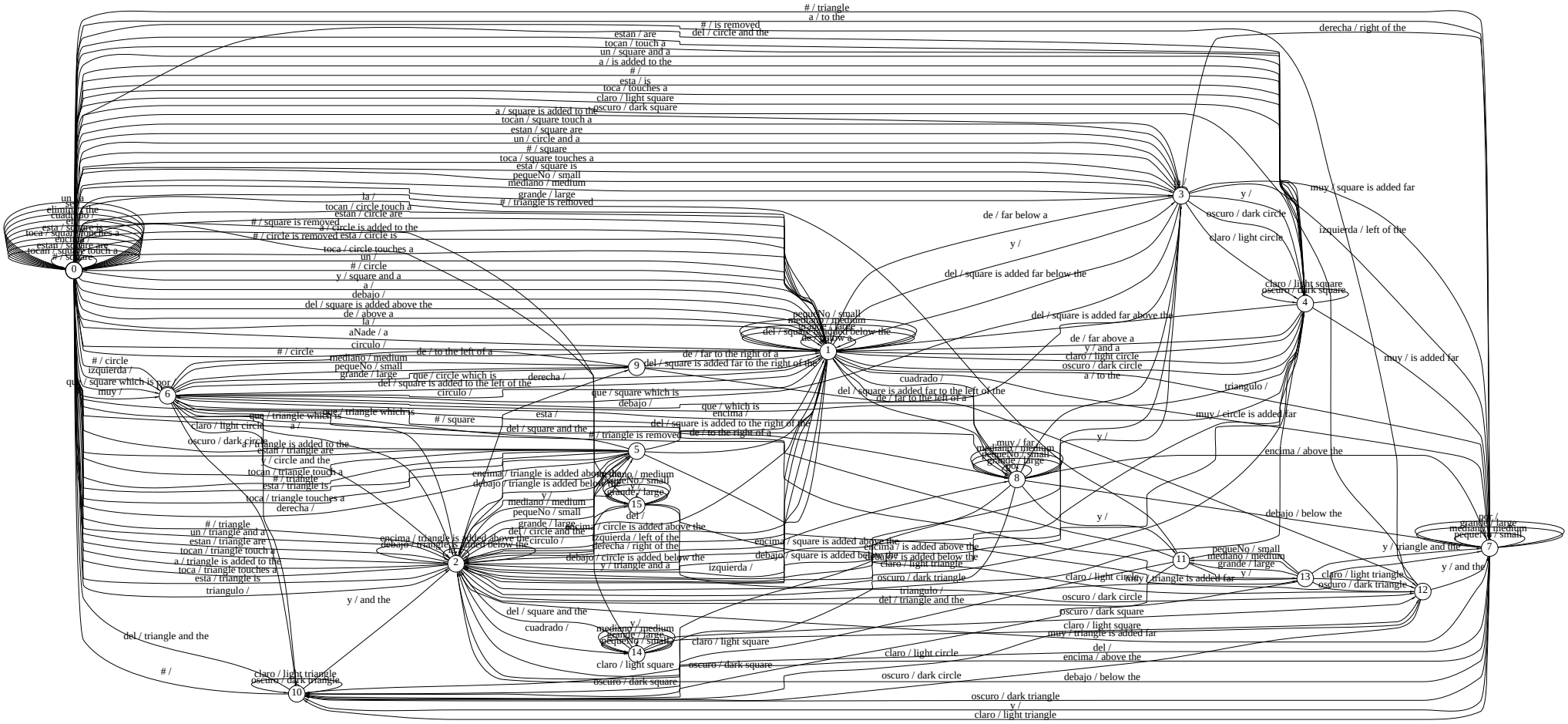
OSTIA learning generalizes the training pairs as much as possible, while preserving the input-output mapping represented by these pairs.
This may lead to *compact* and accurate transducers but they generally involve excessive *over-generalization* of the input and output sentences.

Examples of Spanish sentences accepted (and translated) by a "good" transducer learned by OSTIA (0.0% translation WER for *clean text* input).

debajo izquierda esta por	→	square is removed
elimina un y	→	the a
a y y claro que	→	light square triangle which is
muy esta oscuro	→	dark square which is square

This is *not* a problem for translating *clean text* but it leads to very large translation errors for corrupted text or for *speech input*!

Basic OSTIA–learned SST for Spanish-English MTA



Index

1. Introduction

2. Learning Finite-State Automata

3. Learning Finite-State Transducers

   - Transduction tasks and Models
   - Subsequential Transducer Learning: OSTIA
   - ***Using Input/Output syntactic restrictions: OSTIA–DR***

4. Applications

Helping OSTIA with Domain and/or Range Constraints: OSTIA-DR

Two kind of conditions for OSTIA state merging:

- *Local conditions*: involve only the two states under consideration.
- *Derived merges*: once two states are merged, others may also need to be merged (while possibly "pushing-back" some output substrings) in order to preserve determinism.

LOCAL CONDITIONS:

- *Basic OSTIA*: Allows merging two candidate states if both have the same output or at least one has no output *[Oncina, 91-93]*.
- Additional conditions: use *Finite-State Language Models* (LM) of the Input (or Domain) and/or the Output (or Range) to enforce *Input and/or Output Syntactic Constraints* $\longrightarrow$ two states cannot be merged if they correspond to different states of the Input or Output LMs *[Oncina, 93]*.

The resulting algorithm is known as **OSTIA-DR**

OSTIA-DR

[Oncina,93]

- The use of Domain (and Range) information can be accomplished by labelling each state of the initial Onward Tree Subsequential Transducer (OTSST) with the name of the state of the Domain (or Range) FS Model that would be reached with the corresponding strings.

- The local compatibility rules then include the condition of disallowing the merging of two states if their labels are distinct.

- The resulting SSTs only accept input sentences and only produce output sentences compatible with the syntactic constraints represented by the FSMs used ▷ *This becomes essential for imperfect text or* speech *input.*

- Using OSTIA-DR, the class of *partial* Subsequential Functions can be *indentified in the limit*.

Using Input Language Constraints: OSTIA-D

Algorithm OSTIA-D: //Using Input Language Constraints
INPUT: finite set of input-output pairs, $T \subset (X^* \times Y^*)$
OUTPUT: OST τ consistent with T
τ:=OTST(T); q:=$first(\tau)$;
while q $< last(\tau)$ **do** q:=$next(\tau$,q); q':=$first(\tau)$;
while q' $<$ q **do**
if $\delta_D(p_0, input\_prefix(q')) = \delta_D(p_0, input\_prefix(q))$
and (σ(q') $= \sigma$(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$) **then** τ':=τ;
$merge(\tau$,q',q);
while $\neg subsequential(\tau)$ **do**
let (r,a,v,s), (r,a,v',s') be two edges of τ that
violate the $subsequential$ condition, with s' $<$ s;
if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile endif**
u:=lcp(v',v);
$push\_back(\tau$,u^{-1}v', (r,a,v',s'));
$push\_back(\tau$,u^{-1}v, (r,a,v,s));
if σ(s') $= \sigma$(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
then $merge(\tau$,s',s)
else exitwhile endif
endwhile //$\neg subsequential(\tau)$//
if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile endif**
endif //σ(q') $= \sigma$(q)//
q':=$next(\tau$,q');
endwhile //q' $<$ q//
endwhile //q $< last(\tau)$//
end //OSTIA//

Using Output Language Constraints: OSTIA-R

Algorithm OSTIA-R //Using Output Language Constraints
INPUT: finite set of input-output pairs, $T \subset (X^* \times Y^*)$
OUTPUT: OST τ consistent with T
τ:=OTST(T); q:=$first(\tau)$;
while q $<$ $last(\tau)$ **do** q:=$next(\tau$,q); q':=$first(\tau)$;
while q' $<$ q **do**
if $\delta_R(p_0, output\_prefix(q')) = \delta_R(p_0, output\_prefix(q))$
and (σ(q') $=$ σ(q) **or** σ(q') $= \emptyset$ **or** σ(q) $= \emptyset$) **then** τ':=τ;
$merge(\tau$,q',q);
while $\neg subsequential(\tau)$ **do**
let (r,a,v,s), (r,a,v',s') be two edges of τ that
violate the $subsequential$ condition, with s' $<$ s;
if s' $<$ q **and** v' $\notin Pr$(v) **then exitwhile endif**
u:=lcp(v',v);
$push\_back(\tau$,u^{-1}v', (r,a,v',s'));
$push\_back(\tau$,u^{-1}v, (r,a,v,s));
if σ(s') $=$ σ(s) **or** σ(s') $= \emptyset$ **or** σ(s) $= \emptyset$
then $merge(\tau$,s',s)
else exitwhile endif
endwhile //$\neg subsequential(\tau)$//
if $\neg subsequential(\tau)$ **then** τ:=τ' **else exitwhile endif**
endif //σ(q') $=$ σ(q)//
q':=$next(\tau$,q');
endwhile //q' $<$ q//
endwhile //q $<$ $last(\tau)$//
end //OSTIA//

Spanish-English MTA: OSTIA and OSTIA-DR Learning Results

Translation Word Error Rates for the Extended MTA Feldman's Task, as a function of the Training Set size supplied to OSTIA and OSTIA-DR (with 4-TSS domain Language Model)

Test Set: 10,000 independent input sentences.

Training Set Size	OSTIA WER	OSTIA States	OSTIA Edges	OSTIA-DR WER	OSTIA-DR States	OSTIA-DR Edges
1,000	**58.8%**	412	1652	**55.1%**	813	2023
2,000	**57.0%**	846	3197	**47.1%**	1406	3353
4,000	**51.8%**	1598	5970	**30.1%**	1686	4051
8,000	**3.4%**	186	891	**1.4%**	244	719
16,000	**0.0%**	17	206	**0.0%**	100	363

Using Input/Output syntactic constraints, translation errors can be reduced by a factor of two.

MTA OSTIA and OSTIA-DR Learning: Impact of Noisy Text Input and Input–Output Language Syntactic Constraints

Spanish-English Translation Word Error Rates of distorted test sentences for the Extended MTA Task, as a function of the Training Set size supplied to OSTIA and OSTIA-DR (with 4-TSS Input and Output Language Models). Noisy input Translations obtained using Error-Correcting Parsing.

Test Set: 10,000 **clean** and **5%-distorted** independent input sentences.

Train.Set Size	OSTIA		OSTIA-DR	
	Clean	5%Dist	Clean	5%Dist
8,000	3.4%	**15.0%**	1.4%	**2.7%**
16,000	0.0%	**11.7%**	0.0%	**1.7%**

Using Input/Output syntactic constraints increases robustness dramatically

MTA OSTIA-DR Learning: Examples of distorted input sentences and the obtained translations

I=Original Input; D=5% Distorted Input; T=System Translation.

Correctly Translated:

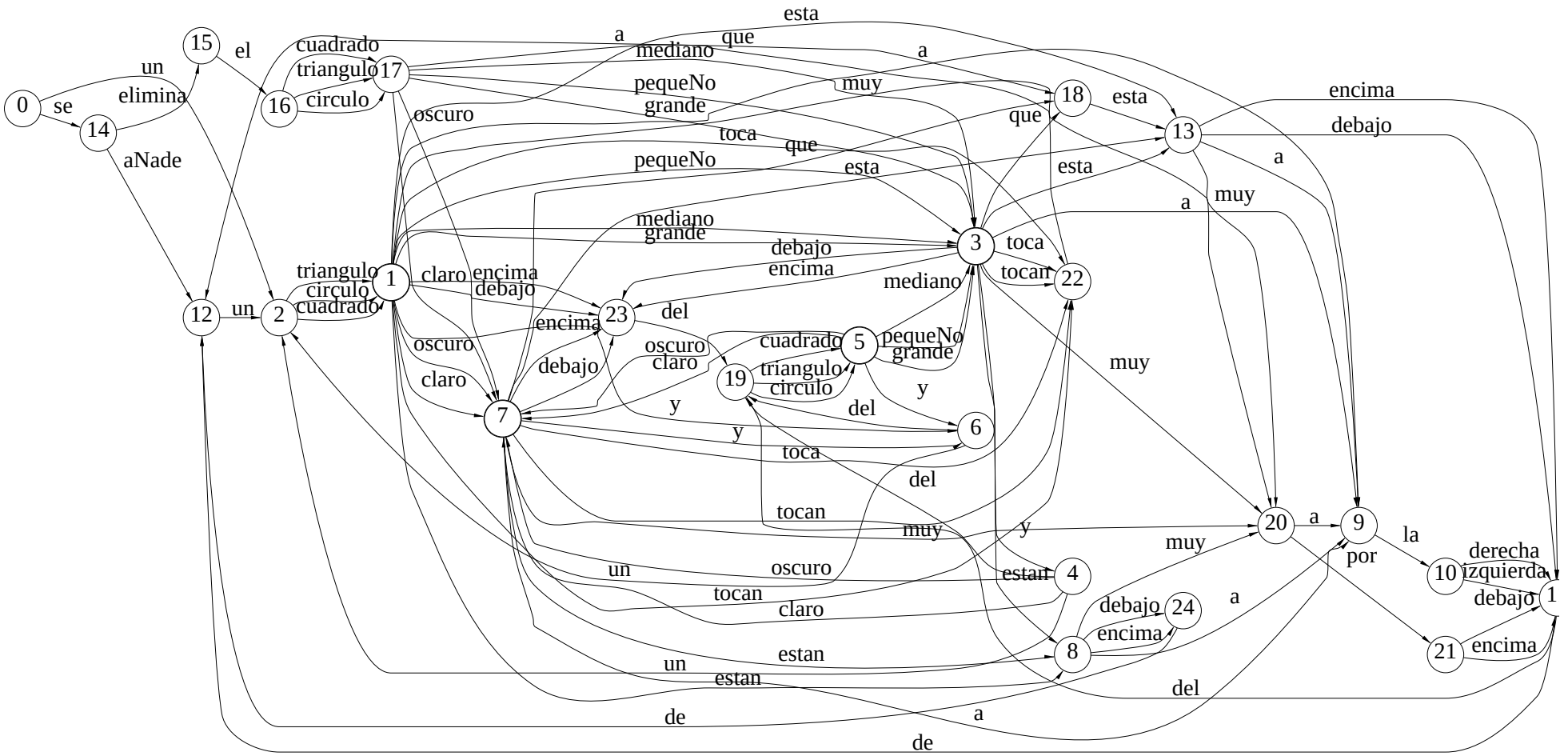
I : *se elimina **el** círculo grande y claro **que** está muy por encima **del** triángulo oscuro y del cuadrado ● mediano*
D: *se elimina **y** círculo grande y claro ● está muy por encima ● triángulo oscuro y del cuadrado **un** mediano*
T: the large light circle which is far above the dark triangle and the medium square is removed
:
I : *un ● círculo mediano y claro **está** debajo de un cuadrado pequeño **y** claro y **un** triángulo pequeño y oscuro*
D: *un **tocan** círculo mediano y claro ● debajo de un cuadrado pequeño claro y **se** triángulo pequeño y oscuro*
T: a medium light circle is below a small light square and a small dark triangle

Translation Errors:

I : *se ● **elimina** el **círculo** que está muy a la izquierda del círculo oscuro y del triángulo mediano y oscuro*
D: *se **de de** el ● que está muy a la izquierda del círculo oscuro y del triángulo mediano y oscuro*
T: the **square** which is far to the left of the dark circle and the medium dark triangle is removed
:
I : *se añade un triángulo **mediano** y claro muy a la derecha del cuadrado mediano y oscuro **y del** círculo pequeño y oscuro*
D: *se añade un triángulo **la** y claro muy a la derecha del cuadrado mediano y oscuro **oscuro claro** círculo pequeño y oscuro*
T: a **small** light triangle is added far to the right of the medium dark square and the small dark circle

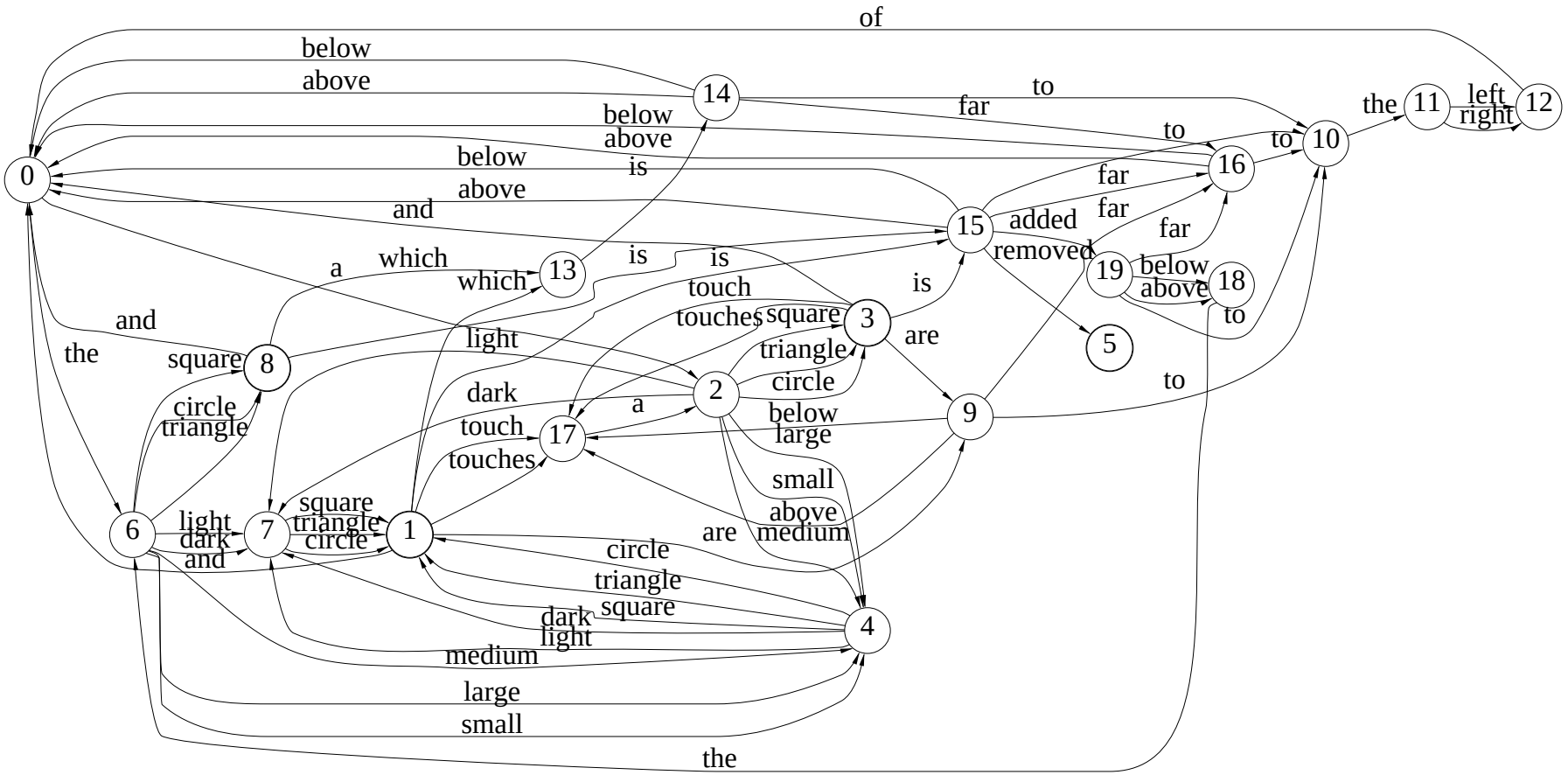
A Finite-State Domain (Spanish) Language Model for MTA

3-TSS Automaton (entailing 3-Gram constraints), learned from 16k Spanish MTA sentences



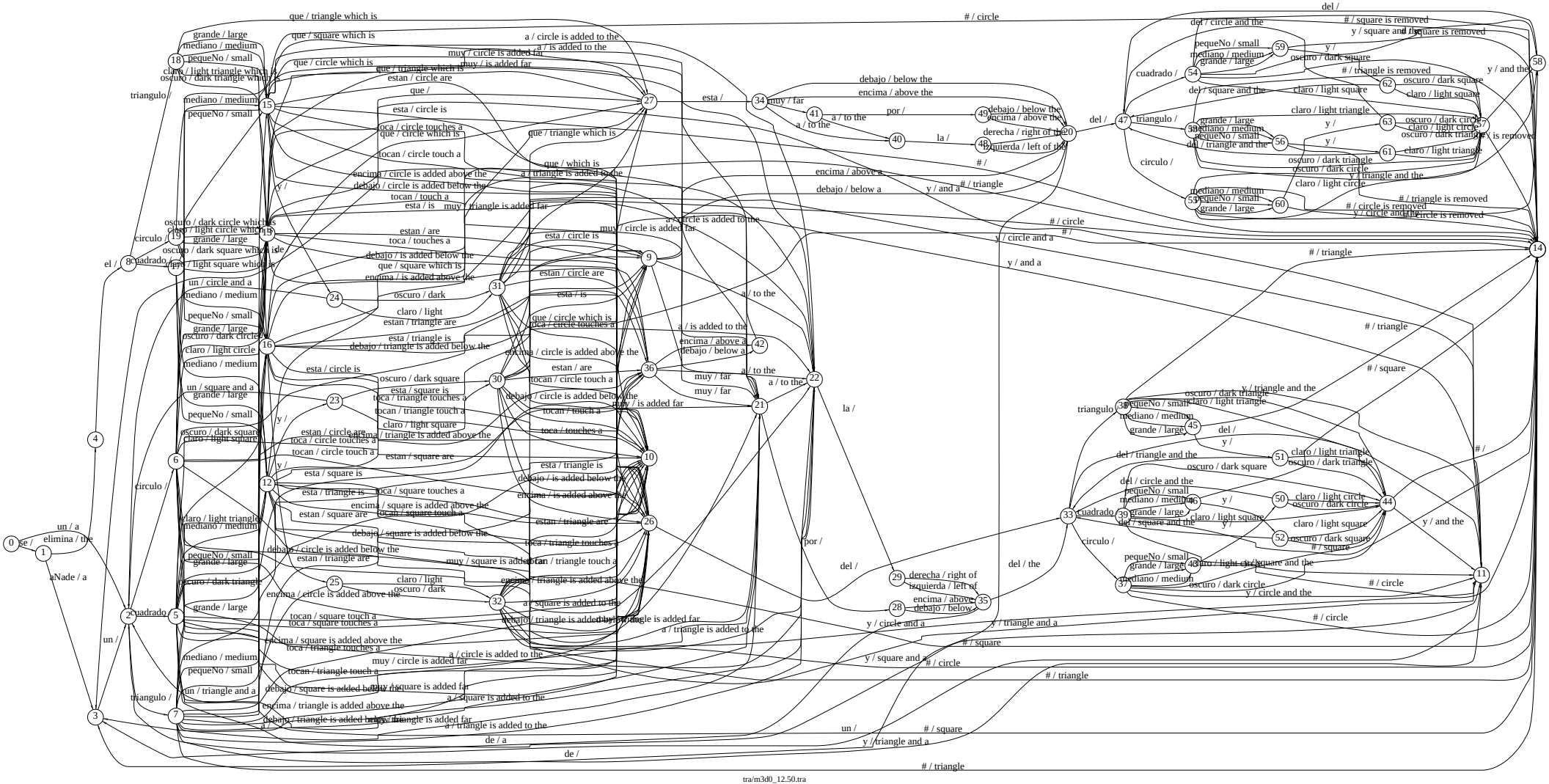
A Finite-State Range (English) Language Model for MTA

3-TSS Automaton (entailing 3-Gram constraints), learned from 16k English MTA sentences



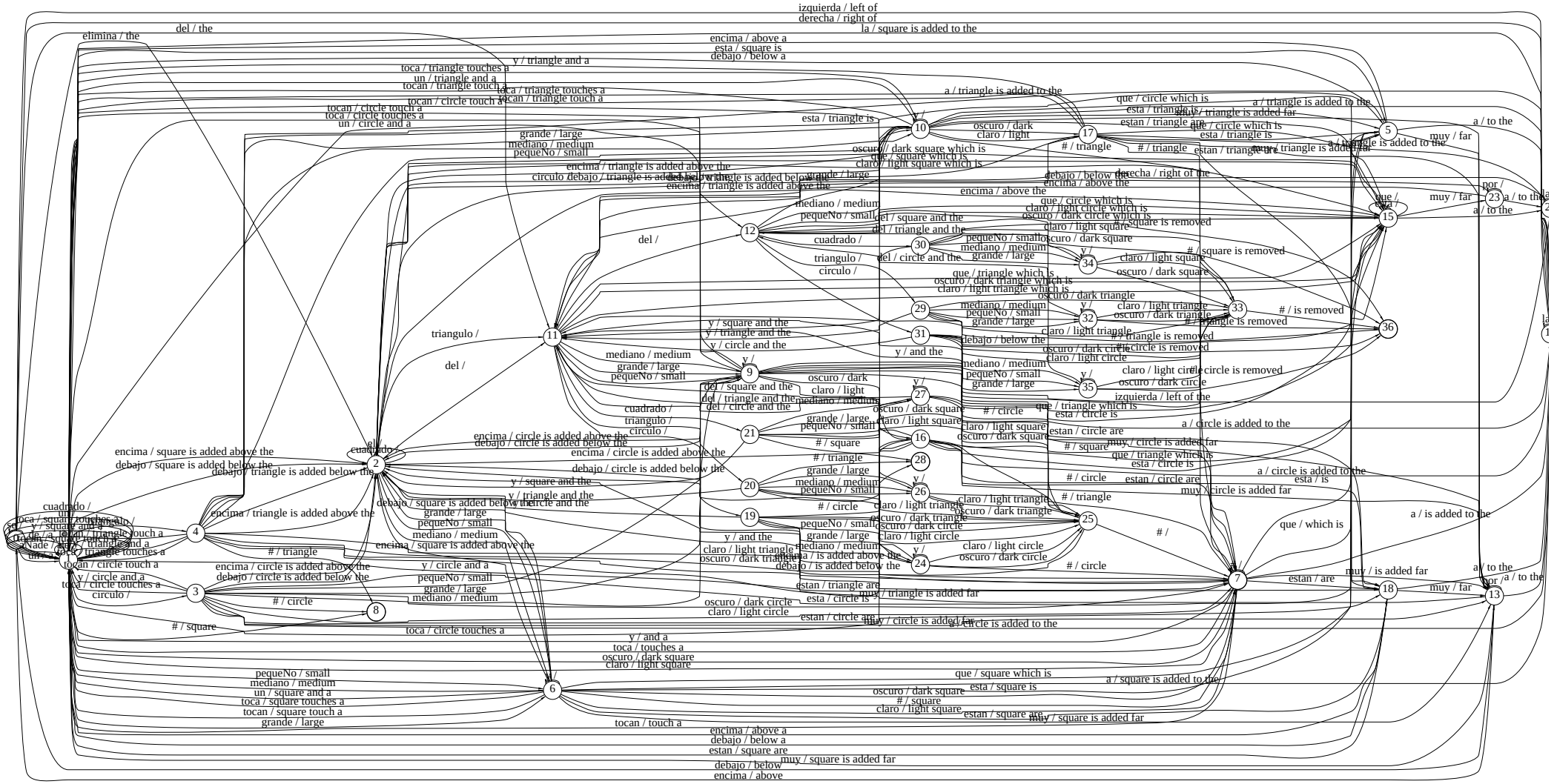
OSTIA-D–learned SST for Spanish-English MTA

learned from 16k I/O MTA pairs, using only *Domain* 3-Gram constraints



OSTIA-R–learned SST for Spanish-English MTA

learned from 16k I/O MTA pairs, using only *Range* 3-Gram constraints



OSTIA-DR–learned SST for Spanish-English MTA

learned from 16k I/O MTA pairs, using both *Domain* and *Range* 3-Gram constraints